

Modeling the Evolution of a Science Project in Software-Reliant System Acquisition Programs

Andrew P. Moore
William E. Novak
Software Engineering Institute
Carnegie Mellon University

International Conference of the System
Dynamics Society

July 24, 2013



Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 24 JUL 2013		2. REPORT TYPE		3. DATES COVERED 00-00-2013 to 00-00-2013	
4. TITLE AND SUBTITLE Modeling the Evolution of a Science Project in Software-Reliant System Acquisition Programs				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Carnegie Mellon University ,Software Engineering Institute,Pittsburgh,PA,15213				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES 31st International Conference of the System Dynamics Society, 21-25 July 2013, Cambridge, MA.					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 21	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

Acquisition Dynamics

The Evolution of a Science Project

9. Warfighters wait years for a new system to be built from scratch.

8. New versions of the system can't be deployed with needed capability, robustness, and performance.

7. New program office unwilling to discard prototype code due to field deployment pressures.

6. Project infrastructure, processes, & staff not able to scale up to production development.

5. As system grows, poor architecture, documentation, & code quality cause poor reliability, performance, & usability.

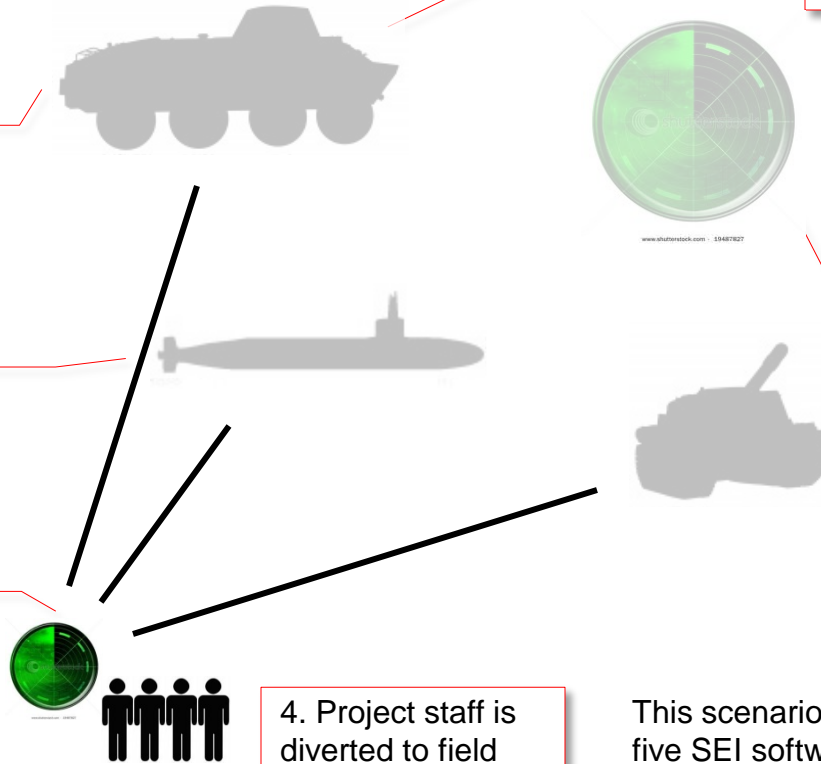
This scenario aggregates five SEI software-reliant system acquisition ITAs conducted in 2006-2009.

4. Project staff is diverted to field support, so development progress slows.

3. Warfighters and field commanders demand more capability, broader deployment, faster response.

2. Prototype is deployed on small scale, and is well received.

1. Project begins as small informal effort to build prototype & prove concept.



Basis for Modeling: Independent Technical Assessments

ITA: Detailed examination of challenged programs with interviews, document reviews, and code analysis

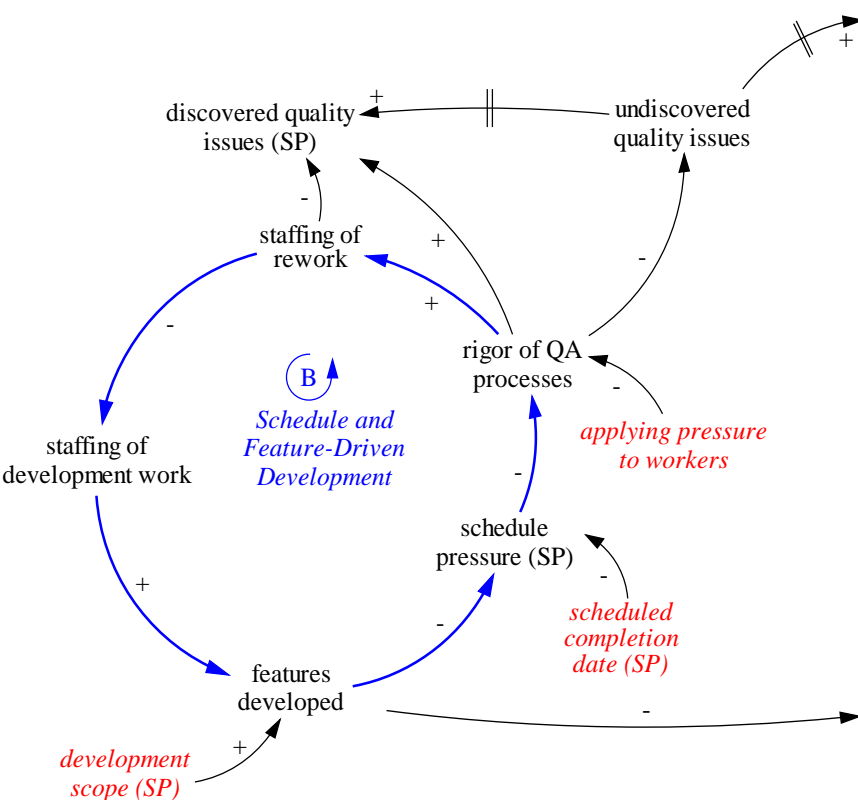
“What they did at first was a proof of concept, a quick and dirty prototype, and when they tried to scale it up, there were indications that it might not be possible...”

—*Acquisition Program Lead*

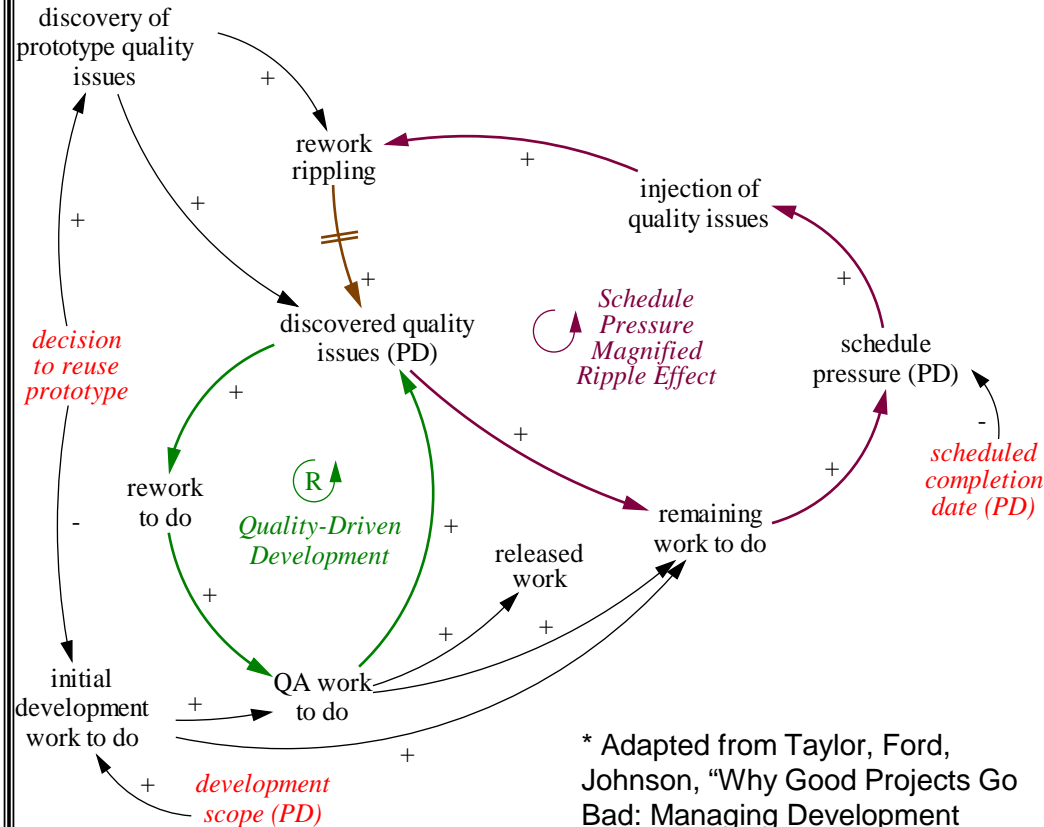


The Evolution of a Science Project

Science Project (SP) Sector



Production Development (PD) Sector *



* Adapted from Taylor, Ford, Johnson, "Why Good Projects Go Bad: Managing Development Projects Near Tipping Points," ICSD 2005, Boston.

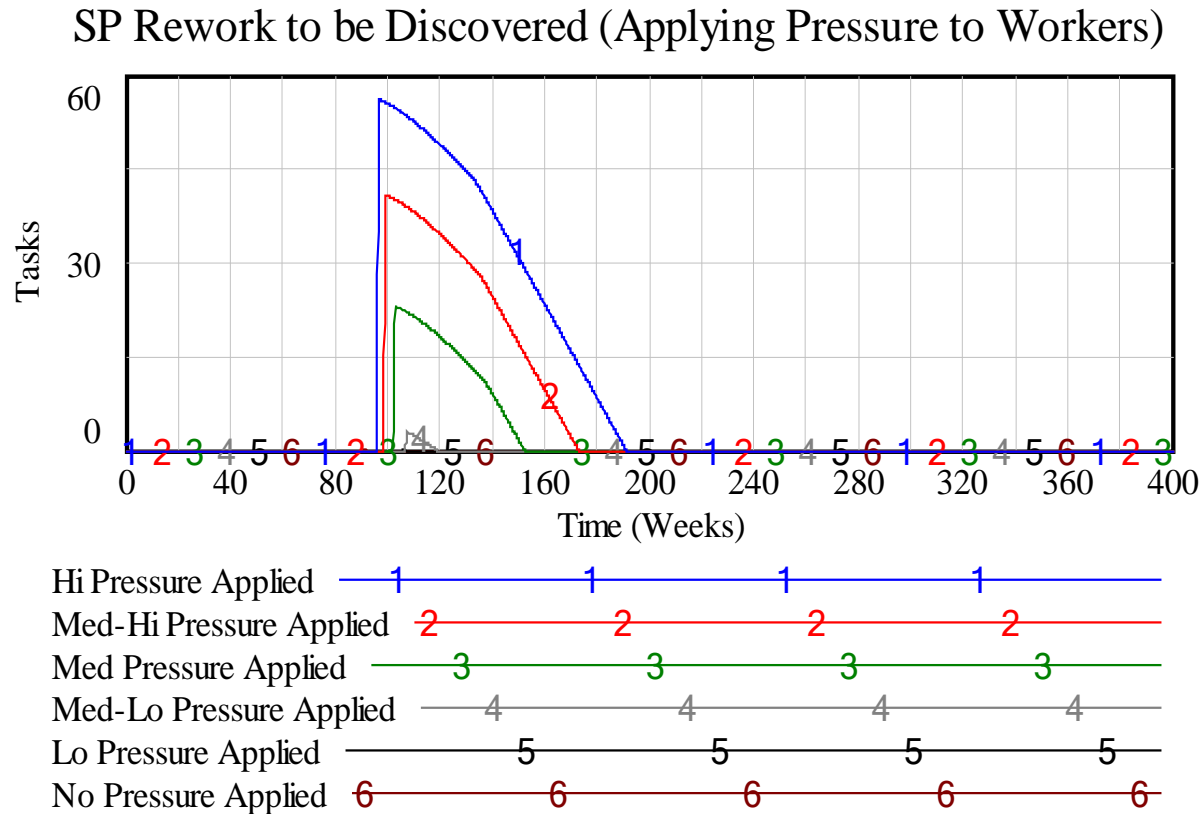


Key Preliminary Findings



The Evolution of a Science Project Assumption

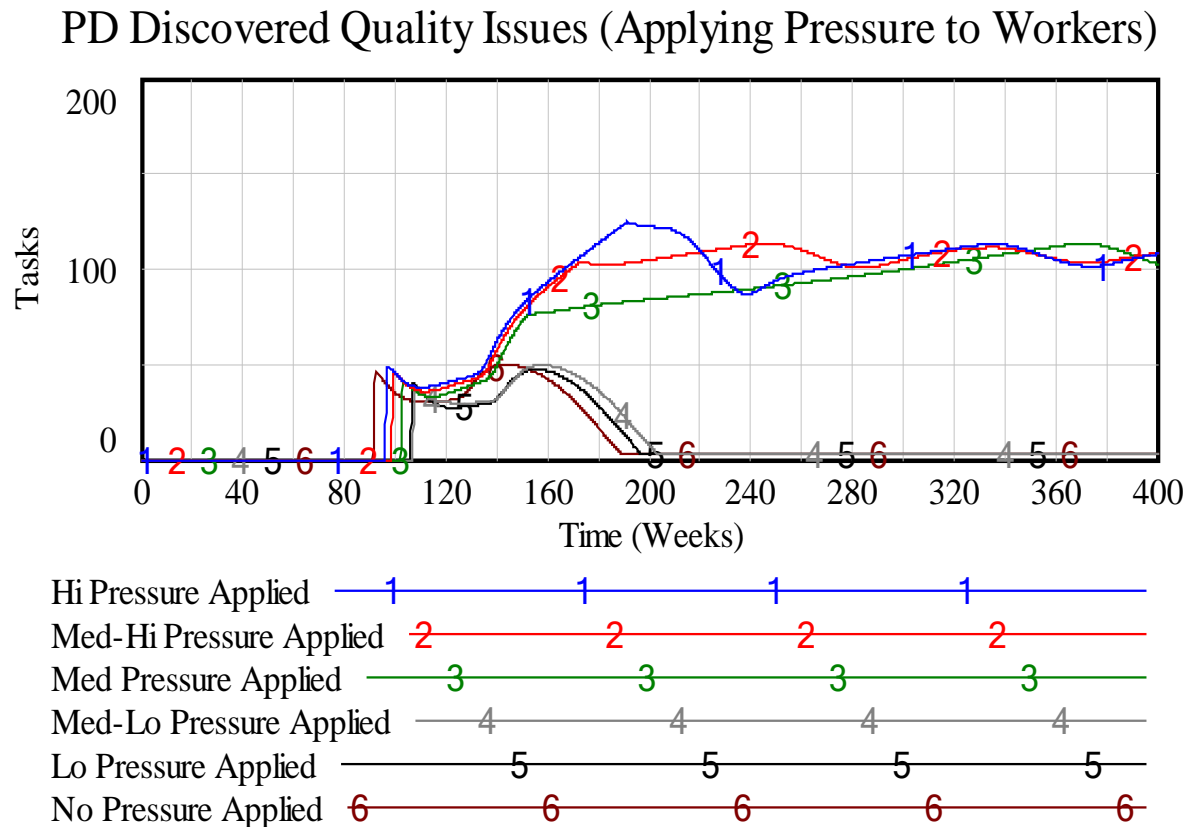
Applying pressure to workers developing SP results in undiscovered rework



The Evolution of a Science Project

Key Preliminary Findings -1

High pressure, or moderate pressure for long periods, can lead to a “tipping point”



The Evolution of a Science Project

The Tipping Point in Evolution of a Science Project

- Accumulating rework creates a dangerous feedback dynamic
- “Firefighting” due to rework is a key underlying element
- Key drivers in reaching the “tipping point” are:
 - a) pressure on developers
 - b) emphasis on schedule and features vs. quality
 - c) timing of the transition from science project to production development
 - d) degree of “ripple effect”



The Evolution of a Science Project

Key Preliminary Findings -2

Placing modest pressure on developers for limited periods shortens schedule

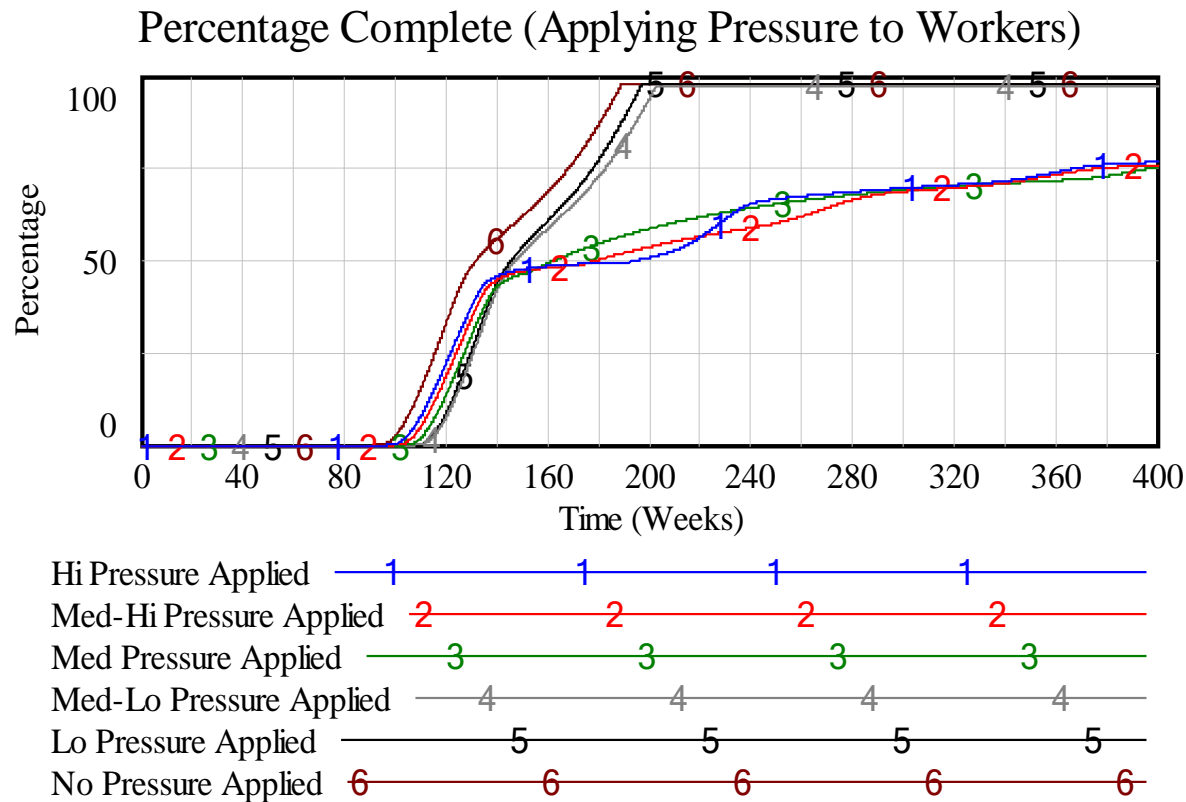
- VenSim optimization shows that placing pressure at a low level is optimal with respect to reducing project duration
- By allowing periods of pressure, followed by periods of relaxation, the program might:
 - Limit worker burnout
 - Perform better regarding schedule



The Evolution of a Science Project

Key Preliminary Findings -3

The tipping point contributes to the “90% Done” Syndrome

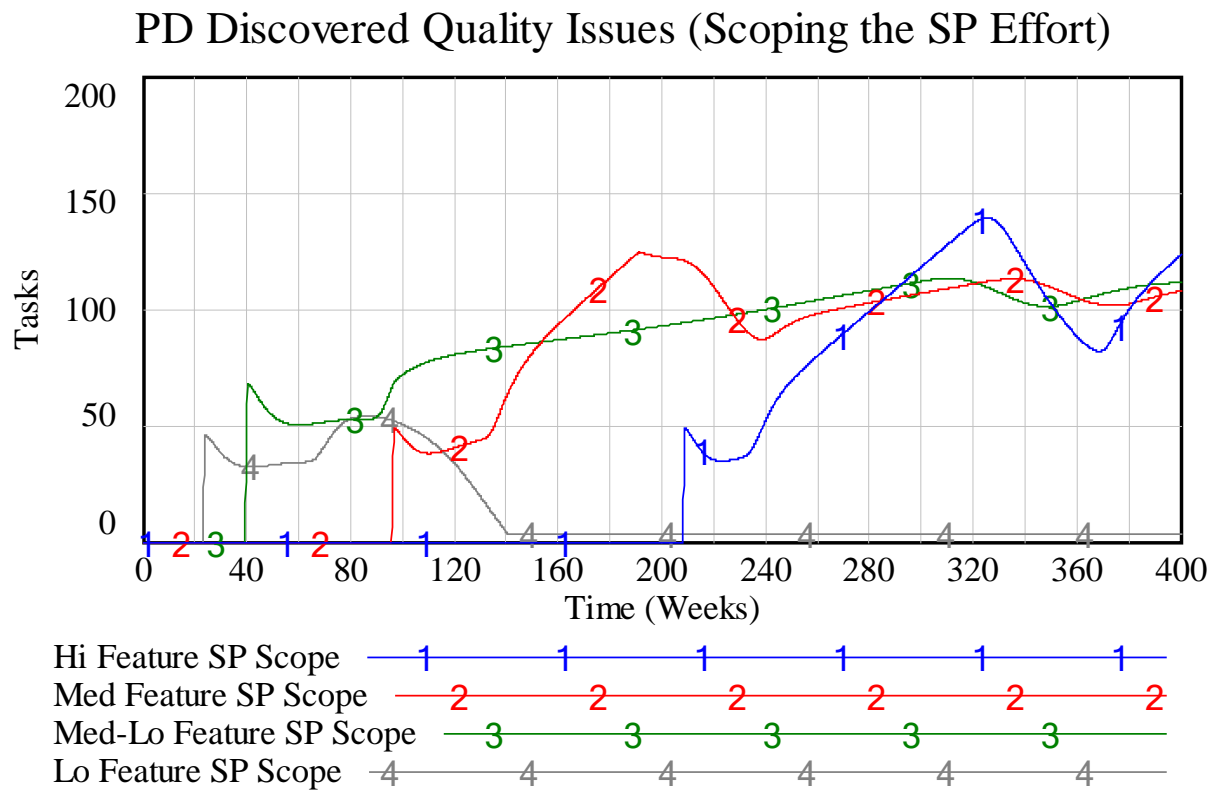


The Evolution of a Science Project

Key Preliminary Findings -4

The transition from science project to production effort should be made *early*

- A late transition increases the amount of undiscovered rework that is transferred

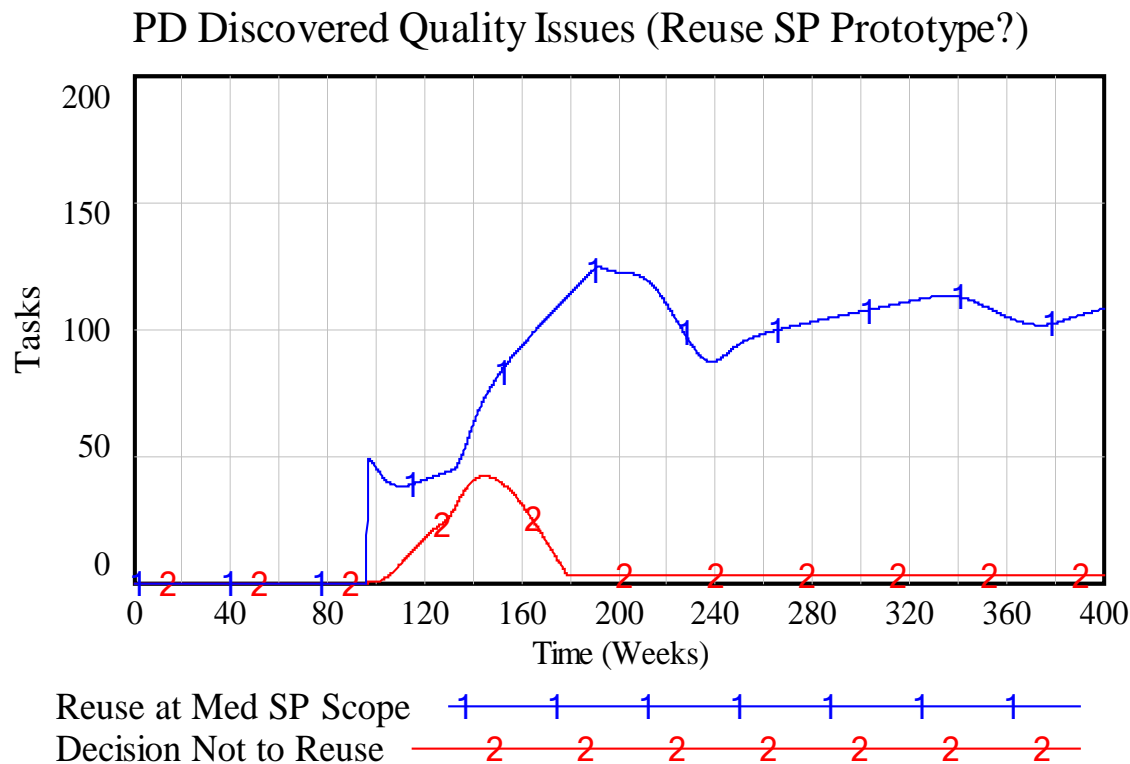


The Evolution of a Science Project

Key Preliminary Findings -5

Throwing away the prototype results in better program performance

- However, very early transition or evolutionary development may also be viable

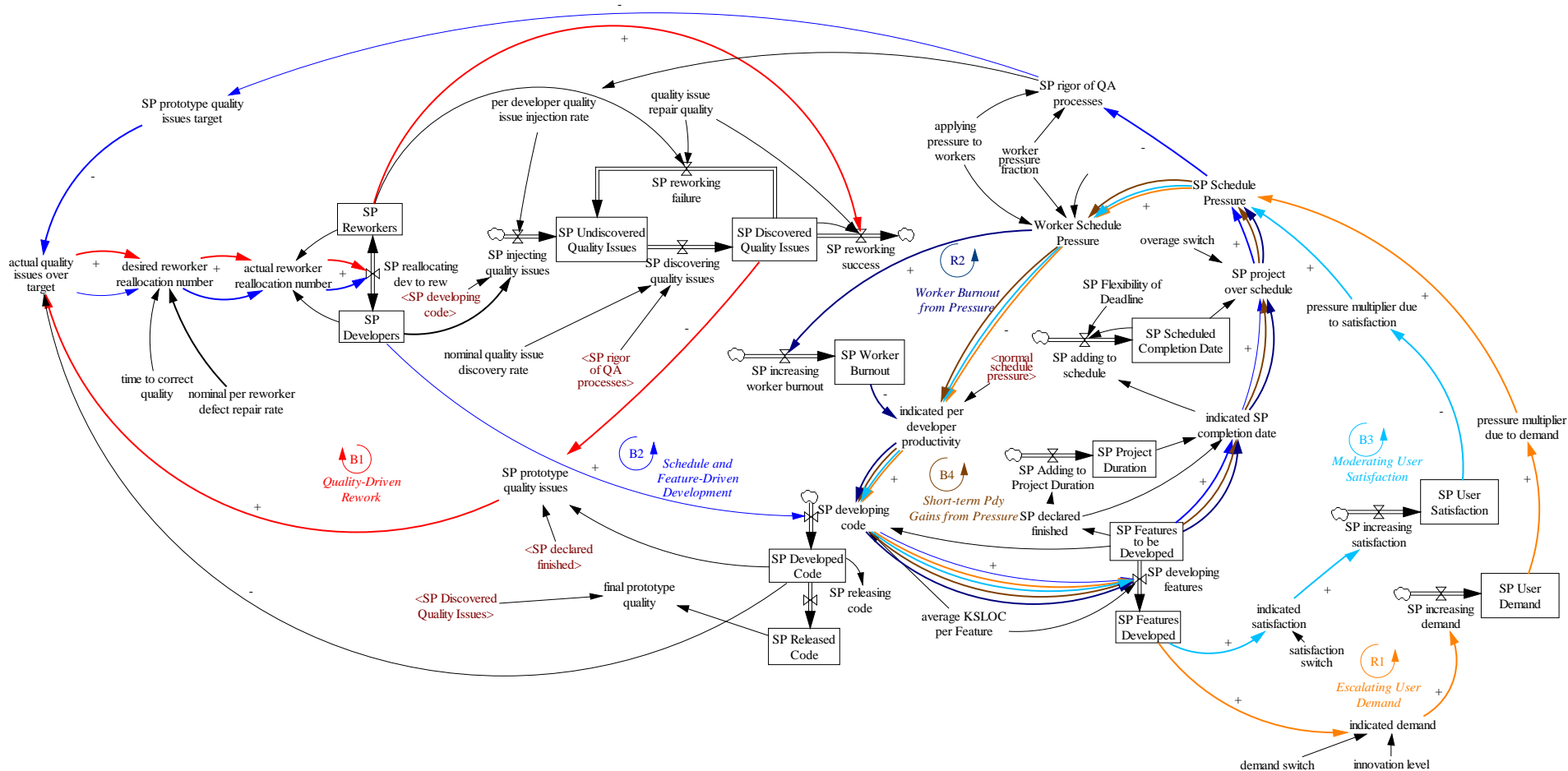




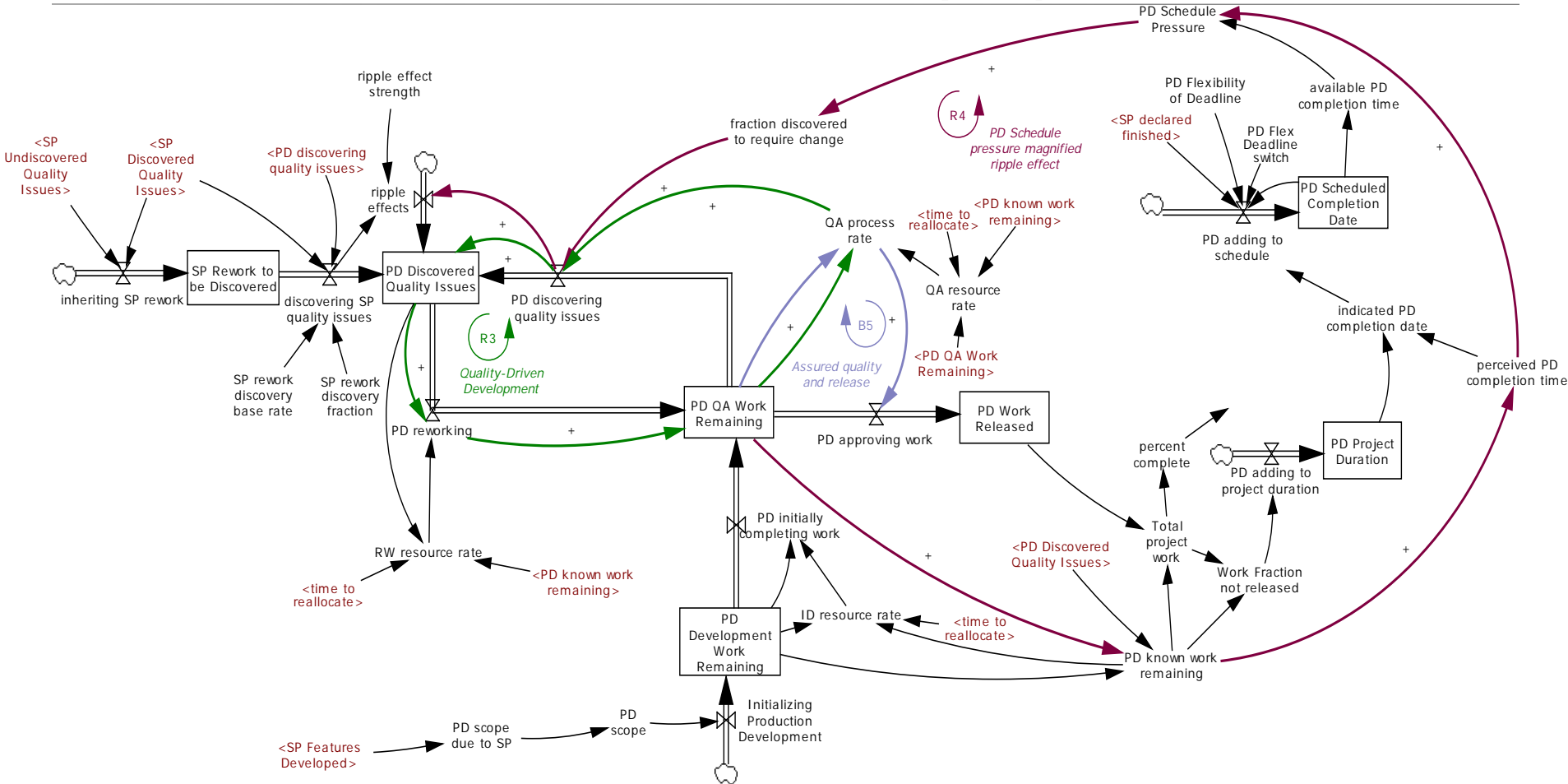
The Simulation Model



The Science Project (SP) Sector



The Production Development (PD) Sector *



* Adapted from the project management models produced by T. Taylor, D. Ford, S. Johnson, "Why Good Projects Go Bad: Managing Development Projects Near Tipping Points, ICSD 2005, Boston.

Preliminary Conclusions

Many acquisition program behaviors are an interaction of the inherent structural dynamics, inc. the incentives that act on key participants.

Tipping points may play a key role in many challenging software development and acquisition situations.

System dynamics modeling can provide a deep understanding of poorly understood dynamics in software-reliant acquisition.

Models are a great source for generating research questions!

Future work:

- Calibrating model based on acquisition program assessments conducted at SEI and data obtained from programs.
- Identifying and evaluating potential mitigations
- Develop games to teach model-based lessons learned



“Firefighting” Animation



Summary and Conclusions

For Additional Information

SEI Report: *"The Evolution of a Science Project: A Preliminary System Dynamics Model of a Recurring Software-Reliant Acquisition Behavior"*

SEI Report: *"Success in Acquisition: Using Archetypes to Beat the Odds"*

SEI Blog: *"Themes Across Acquisition Programs": Parts 1-4*

Website: <http://www.sei.cmu.edu/acquisition/research/archetypes.cfm>

Download all twelve:

- PMO vs. Contractor Hostility
- Underbidding the Contract
- Everything for Everybody
- The Bow Wave Effect
- Brooks' Law
- Firefighting
- "Happy Path" Testing
- Longer Begets Bigger
- Shooting the Messenger
- Feeding the Sacred Cow
- Staff Burnout and Turnover
- Robbing Peter to Pay Paul



Summary and Conclusions

Joint Program Acquisition Experience Wanted!

We are analyzing the dynamic organizational behavior of joint and joint-interest programs as part of an ongoing research project.

We are conducting group modeling workshops to elicit key joint program behaviors, and are using the information to build a system dynamics model.

If you'd be interested in participating in a workshop, or collaborating with us in other ways, please contact:

William E. Novak
Senior Member of Engineering Staff
Office: 412.268.5519
Email: wen@sei.cmu.edu
Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213-3890



This material is based upon work supported by the U.S. Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY

THIS MATERIAL OF CARNEGIE MELLON UNIVERSITY AND ITS SOFTWARE ENGINEERING INSTITUTE IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.





Software Engineering Institute

Carnegie Mellon



Software Engineering Institute

Carnegie Mellon

Software Technology Conference
April 10, 2013
© 2013 Carnegie Mellon University